Appl. Comput. Harmon. Anal. 32 (2012) 145–154

Contents lists available at SciVerse ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Letter to the Editor

Split-Bregman iteration for framelet based image inpainting ☆

Qia Li [a], Lixin Shen [b,a,*,1], Lihua Yang [a]

[a] School of Mathematics and Computational Sciences, Sun Yat-sen University, Guangzhou 510275, PR China
[b] Department of Mathematics, Syracuse University, Syracuse, NY 13244, United States

ARTICLE INFO

Article history:
Received 24 April 2011
Revised 16 August 2011
Accepted 27 September 2011
Available online 1 October 2011
Communicated by Charles K. Chui

Keywords:
Bregman iteration
Framelet
Inpainting

ABSTRACT

Image inpainting plays a significant role in image processing and has many applications. Framelet based inpainting methods were introduced recently by Cai et al. (2007, 2009) [6,7,9] under an assumption that images can be sparsely approximated in the framelet domain. By analyzing these methods, we present a framelet based inpainting model in which the cost functional is the weighted ℓ^1 norm of the framelet coefficients of the underlying image. The split-Bregman iteration is exploited to derive an iterative algorithm for the model. The resulting algorithm assimilates advantages while avoiding limitations of the framelet based inpainting approaches in Cai et al. (2007, 2009) [6,7,9]. The convergence analysis of the proposed algorithm is presented. Our numerical experiments show that the algorithm proposed here performs favorably.

1. Introduction

Image inpainting refers to the filling-in of missing data in digital images based on the information available in the observed region. The notion of image inpainting was first introduced into digital image processing in [1]. Since then, broad applications of image inpainting have been found in image processing such as image restoration, film restoration, text or scratch removal, error concealment, and digital zooming [2,10,12–14].

The problem of image inpainting can be formulated as follows. We view an image as a vector in $\mathbb{R}^n$ by sequentially concatenating the columns of the image. Let f be an original image defined on the image domain $\Omega := \{1, 2, \ldots, n\}$. Let Λ be a nonempty subset of Ω where original information of the image f is available in an observed image g. That is, original information of the image on $\Omega \setminus \Lambda$ is missing. Let P_Λ be an $n \times n$ diagonal matrix whose i-th diagonal entry is 1 if $i \in \Lambda$ and 0 otherwise. With this matrix, our goal is to find f from g such that f interpolates g on Λ, i.e.,

$$P_\Lambda f = P_\Lambda g. \tag{1}$$

Many successful inpainting methods have been proposed. Most of them are based on numerical PDEs [1,2,10–13]. Recently, inpainting algorithms based on redundant systems have been developed and their efficiency for image inpainting has been also demonstrated, see, for example, [6,7,9,14]. We are particularly interested in algorithms with redundant systems generated by framelet systems. Motivations for developing framelet based algorithms are that (i) the redundance of a framelet system allows us to squeeze the information from the given data on Λ to the missing data on $\Omega \setminus \Lambda$ by perturbing

☆ This research is supported in part by the US National Science Foundation under grants DMS-0712827 and DMS-1115523, by Guangdong Provincial Government of China through the "Computational Science Innovative Research Team" program, and by Guangdong Province Key Lab of Computational Science.

* Corresponding author at: Department of Mathematics, Syracuse University, Syracuse, NY 13244, United States.
E-mail address: lshen03@syr.edu (L. Shen).
[1] All correspondence should be sent to this author.

1063-5203/$ – see front matter
doi:10.1016/j.acha.2011.09.007

the framelet coefficients of the observed image; (ii) many images (e.g. cartoon images) can be modeled as piecewise smooth functions and framelets can provide a good approximation to such functions via nonlinear approximation (see e.g. [3]); and (iii) the redundant system reduces artifacts caused by data filling schemes. In what follows, the framelet decomposition operator associated with a tight framelet system is represented by an $m \times n$ matrix A with $m > n$ such that $A^\top A = I$, where I is the $n \times n$ identity matrix. Thus, for every vector $u \in \mathbb{R}^n$, $u = A^\top(Au)$, where the components of Au are called the framelet coefficients of u.

Our inpainting algorithm proposed in this paper is motivated by the framelet based inpainting algorithm (FIA) in [6,7] and the framelet and Bregman based inpainting algorithm (FBIA) in [9]. To better present the motivation of our inpainting algorithm, we give a brief review to the FIA and FBIA for the model (1).

The framelet based inpainting algorithm (FIA) roughly has four steps. First, the image to be inpainted is transformed into a transform domain using a framelet system. Secondly, the framelet coefficients are modified by a thresholding operator with a carefully chosen threshold. Thirdly, the transformation is inverted to obtain an output image. Finally, an inpainted image is obtained from the output image by replacing those pixels in Λ with the corresponding pixels of the image to be inpainted. These four steps can be iteratively applied until certain stopping criteria is met. The corresponding FIA is as follows:

$$f^{k+1} = (I - P_\Lambda)A^\top \mathcal{T}_\lambda\big(Af^k\big) + P_\Lambda g, \quad k = 0, 1, \ldots, \tag{2}$$

where λ is a vector in $\mathbb{R}^m$ with nonnegative components and $\mathcal{T}_\lambda$ is the soft thresholding operator

$$\mathcal{T}_\lambda(x) := \big(t_{\lambda_1}(x_1), t_{\lambda_2}(x_2), \ldots, t_{\lambda_m}(x_m)\big)^\top \tag{3}$$

with $t_{\lambda_i}(x_i) = \max\{|x_i| - \lambda_i, 0\}\frac{x_i}{|x_i|}$. It was further shown in [6] that the resulting sequence $\{f^k\}$ converges to a solution of the following constrained minimization problem

$$\min_{f,d}\left\{\big\|\operatorname{diag}(\lambda)d\big\|_1 + \frac{1}{2}\|Af - d\|_2^2 \colon\ f \in \mathbb{R}^n,\ d \in \mathbb{R}^m,\ P_\Lambda f = P_\Lambda g\right\}. \tag{4}$$

Selecting the vector λ in (2) and (4) is crucial. The iteration (2) with λ having large components will damage small features of the underlying image while this iteration with λ having small components will fail to remove the artifacts of the underlying image. To take these observations into consideration, it was suggested in [7] that for a given λ having sufficiently large components, this vector in (2) is reduced by half every fixed number of iterations until the procedure converges.

The framelet and Bregman based inpainting algorithm (FBIA) was proposed for the minimization problem

$$\min_f\big\{\|Af\|_1 \colon\ f \in \mathbb{R}^n,\ P_\Lambda f = P_\Lambda g\big\}. \tag{5}$$

The idea of Bregman iteration is to transfer the constrained problem (5) into a series of the following unconstrained ones

$$\begin{cases} f^{k+1} = \arg\min_f\left\{\|Af\|_1 + \dfrac{\mu}{2}\big\|P_\Lambda f - P_\Lambda g + c^k\big\|_2^2 \colon\ f \in \mathbb{R}^n\right\}, \\ c^{k+1} = c^k + \big(P_\Lambda f^{k+1} - P_\Lambda g\big), \end{cases} \tag{6}$$

where $c^0 = 0$ and $\mu > 0$ is a parameter introduced by the Bregman iteration. It was proposed in [9] to apply the split-Bregman technique [15] for solving f^{k+1} in the first step of (6). The corresponding algorithm is

$$\begin{cases} u^{n+1} = \arg\min_u\left\{\dfrac{\mu}{2}\big\|P_\Lambda u - P_\Lambda g + c^k\big\|_2^2 + \dfrac{\gamma}{2}\big\|Au - d^n + b^n\big\|_2^2 \colon\ u \in \mathbb{R}^n\right\}, \\ d^{n+1} = \arg\min_d\left\{\|d\|_1 + \dfrac{\gamma}{2}\big\|d - Au^{n+1} - b^n\big\|_2^2 \colon\ d \in \mathbb{R}^m\right\}, \\ b^{n+1} = b^n + \big(Au^{n+1} - d^{n+1}\big), \end{cases} \tag{7}$$

where $b^0 = 0$, $d^0 = 0$, $\gamma > 0$ is a parameter introduced by the split-Bregman method. The limit of u^n in (7) is considered to be f^{k+1} in (6).

By comparing the FIA with the FBIA, the advantages of the FIA and the corresponding model (4) over the FBIA and the corresponding model (5) are that the vector λ in model (4) provides some flexibility for handling framelet coefficients and the pixels of f^k on Λ are always consistent with the available data of g in the FIA. On the other hand, the advantage of the FBIA and the corresponding model (5) over the FIA and the corresponding model (4) is that the parameters μ and γ in the FBIA remain unchanged, however, the vector λ in the FIA varies.

Motivated by the FIA and the FBIA, we consider the following framelet based minimization model

$$\min_f\big\{\big\|\operatorname{diag}(\lambda)Af\big\|_1 \colon\ f \in \mathbb{R}^n,\ P_\Lambda f = P_\Lambda g\big\} \tag{8}$$

for image inpainting and develop its numerical algorithm by using the split-Bregman technique. Our proposed algorithm will possess the advantages of both the FIA and the FBIA. That is, the parameters in the algorithm are fixed and the output

of each iterate of the algorithm matches with the observed data on Λ. Moreover, unlike the FBIA in which the Bregman iteration technique was used twice, our algorithm uses the Bregman iteration technique only once. As demonstrated in numerical experiments, our algorithm converges in relatively less number of iterations than the FBIA.

The remaining part of the paper is organized as follows. In Section 2, we propose our framelet based split-Bregman iteration inpainting algorithm. In Section 3, we present the convergence analysis of the algorithm. Section 4 is devoted to numerical experiment of the algorithm in the context of impulsive noise removal and compares the numerical performance of our proposed algorithm with that of the FIA and FBIA. Finally, our conclusions are given in Section 5.

2. A framelet based split-Bregman iteration inpainting algorithm

In this section, we propose our framelet based split Bregman iteration inpainting algorithm. We begin by reviewing the Bregman iteration technique.

The Bregman iteration method is originated in functional analysis for finding extremum of a convex function [4]. It has been successfully applied to improve image reconstruction quality in the context of total-variation regularization [18] and to solve ℓ_1-related minimization problems [8,9,15,20]. The Bregman iteration is defined via the Bregman distance. For a convex function J, the Bregman distance with respect to J between two points u and v is defined as

$$B_J^p(u, v) := J(u) - J(v) - \langle u - v, p\rangle,$$

where p is in $\partial J(v) := \{q\colon\ J(s) \geqslant J(v) + \langle s - v, q\rangle,\ \text{for all } s\}$ the subdifferential of J at the point v. The $B_J^p(u, v)$ measures the closeness between u and v in the sense that $B_J^p(u, v) \geqslant 0$ and $B_J^p(u, v) \geqslant B_J^p(w, v)$ for all points w on the line segment connecting u and v.

Let $L \in \mathbb{R}^{m\times n}$ and $g \in \mathbb{R}^m$ be given. We briefly review the Bregman iteration for the following constrained minimization problem

$$\min_u \big\{ J(u)\colon\ u \in \mathbb{R}^n,\ Lu = g \big\}, \tag{9}$$

where J is a convex function. We assume that $J(u)$ is coercive, i.e., whenever $\|u\|_2 \to \infty$, $J(u) \to \infty$. This ensures that the set of solutions to problem (9) is nonempty.

Given $u^0 = p^0 = 0$, the Bregman iteration for problem (9) is generated by

$$\begin{cases} u^{k+1} = \arg\min_u \left\{ B_J^{p^k}\big(u, u^k\big) + \dfrac{\gamma}{2}\|Lu - g\|_2^2\colon\ u \in \mathbb{R}^n \right\}, \\ p^{k+1} = p^k - \gamma L^\top\big(Lu^{k+1} - g\big) \in \partial J\big(u^{k+1}\big), \end{cases} \tag{10}$$

where γ is a fixed step size. The convergence analysis of the Bregman iteration was given in [18]. In particular, under fairly weak assumptions on J, it was proved that $\|Lu^k - g\|_2 \to 0$ as $k \to \infty$. The Bregman iteration (10) can be reformulated into a compact form as follows: Given $b^0 = 0$,

$$\begin{cases} u^{k+1} = \arg\min_u \left\{ J(u) + \dfrac{\gamma}{2}\big\|Lu - g + b^k\big\|_2^2 \right\}, \\ b^{k+1} = b^k + \big(Lu^{k+1} - g\big). \end{cases} \tag{11}$$

This form will be used in the remaining part of this paper. As an example, Eq. (6) is the Bregman iteration for problem (5).

Next, we will present our framelet based split-Bregman iteration algorithm for problem (8). To this end, we introduce some notations. For a vector $f \in \mathbb{R}^n$, $f_{|\Omega\setminus\Lambda}$ and $f_{|\Lambda}$ are vectors formed by the components of f corresponding to the indices in $\Omega\setminus\Lambda$ and Λ, respectively. For the framelet matrix A, $A_{|\Omega\setminus\Lambda}$ and $A_{|\Lambda}$ are matrices formed by the columns of A corresponding to the indices in $\Omega\setminus\Lambda$ and Λ, respectively. Then we have that

$$Af = A_{|\Omega\setminus\Lambda} f_{|\Omega\setminus\Lambda} + A_{|\Lambda} f_{|\Lambda}.$$

For simplicity, for problem (8), we denote

$$u := f_{|\Omega\setminus\Lambda}, \qquad D := A_{|\Omega\setminus\Lambda}, \quad \text{and} \quad c := A_{|\Lambda} f_{|\Lambda}.$$

Further, the constraint $P_\Lambda f = P_\Lambda g$ yields $f_{|\Lambda} = g_{|\Lambda}$, that is, the vector c is available. Hence, finding f in problem (8) is equivalent to finding u in the following minimization problem

$$\min_u \big\{ \big\|\mathrm{diag}(\lambda)(Du + c)\big\|_1\colon\ u \in \mathbb{R}^{|\Omega\setminus\Lambda|} \big\}, \tag{12}$$

where $|\Omega \setminus \Lambda|$ is the cardinality of the set $\Omega \setminus \Lambda$.

To develop the Bregman iteration method for problem (12), we introduce an auxiliary variable $d = Du + c$, whereby problem (12) becomes

$$\min_{u,d} \big\{ \big\|\mathrm{diag}(\lambda)d\big\|_1\colon\ u \in \mathbb{R}^{|\Omega\setminus\Lambda|},\ d \in \mathbb{R}^m,\ Du + c = d \big\}. \tag{13}$$

By identifying (u,d), $\|\mathrm{diag}(\lambda)d\|_1$, $Du-d$, and $-c$ as u, $J(u)$, Lu, and g, respectively, in problem (9), and following the iterative procedure (11), the Bregman iteration method for problem (12) is

$$\begin{cases} \big(u^{k+1},d^{k+1}\big)=\arg\min\limits_{u,d}\left\{\big\|\mathrm{diag}(\lambda)d\big\|_1+\dfrac{\gamma}{2}\big\|Du+c-d+b^k\big\|_2^2\colon\ u\in\mathbb{R}^{|\Omega\setminus\Lambda|},\ d\in\mathbb{R}^m\right\},\\ b^{k+1}=b^k+\big(Du^{k+1}+c-d^{k+1}\big), \end{cases} \tag{14}$$

for $k=0,1,\ldots,$ starting with $b^0=0$. The convergence of $\{u^k\}$ and $\{d^k\}$ from (14) was discussed in [17,18] under the assumption that the minimization problem in the first step of (14) was solved exactly. An approach proceeds by alternatively updating u^{k+1} and d^{k+1}, eventually converging to an optimum of minimization problem (14). This approach is described as follows:

$$\begin{cases} \text{Input: } b^k,\ c,\ u^k=0\\ \text{while ``}u^k\text{ and }d^k\text{ do not converge'' do}\\ \quad d^{k+1}\leftarrow\arg\min\limits_{d}\left\{\big\|\mathrm{diag}(\lambda)d\big\|_1+\dfrac{\gamma}{2}\big\|Du^{new}+c-d+b^k\big\|_2^2\colon\ d\in\mathbb{R}^m\right\},\\ \quad u^{k+1}\leftarrow\arg\min\limits_{u}\left\{\dfrac{\gamma}{2}\big\|Du+c-d^{new}+b^k\big\|_2^2\colon\ u\in\mathbb{R}^{|\Omega\setminus\Lambda|}\right\},\\ \text{end} \end{cases} \tag{15}$$

where d^{new} (respectively u^{new}) is either d^{k+1} (respectively u^{k+1}) if it is available or d^k (respectively u^k) otherwise. Updating u^{k+1} and d^{k+1} can be efficiently and explicitly computed. In fact,

$$d^{k+1}=\mathcal{T}_{\lambda/\gamma}\big(Du^{new}+b^k+c\big)\quad\text{and}\quad u^{k+1}=D^\top\big(d^{new}-b^k\big),$$

due to $D^\top D=I$ and $D^\top c=0$. However, it is not desirable to solve the minimization problem in (15) to full convergence. As numerically tested in [9,15], it is usually enough to update u^{k+1} and d^{k+1} once by the approach (15). We therefore propose a three-step split-Bregman iteration method for problem (12). This method is summarized in Algorithm 1.

input : observed image $g\in\mathbb{R}^n$, set $\Lambda\subset\Omega:=\{1,\ldots,n\}$ on which original information of the underlying image is available, vector $\lambda\in\mathbb{R}^m$ with nonnegative components, regularization parameter $\gamma>0$, step size $0<\delta\leqslant 1$, framelet matrix $A\in\mathbb{R}^{m\times n}$
output: image $f^\star\in\mathbb{R}^n$
Initialize: $k=0$, $b^0=0$, $u^0=g_{|\Omega/\Lambda}$, $c=A_{|\Lambda}g_{|\Lambda}$
while *"u^k do not converge"* **do**
  Step 1: $d^{k+1}\leftarrow\mathcal{T}_{\lambda/\gamma}(Du^k+c+b^k)$
  Step 2: $u^{k+1}\leftarrow D^\top(d^{k+1}-b^k)$
  Step 3: $b^{k+1}\leftarrow b^k+\delta(Du^{k+1}+c-d^{k+1})$
end
Write the output of u^k of the above loop as u^∞, let $f^\star_{|\Lambda}:=g_{|\Lambda}$ and $f^\star_{|\Omega/\Lambda}:=u^\infty$

Algorithm 1: Framelet based split-Bregman iteration inpainting algorithm.

In Algorithm 1, the parameters λ, γ, δ are fixed and the pixel values of the solution to problem (12) are the essentially same as those of g on the set Λ in each iteration. In this sense, Algorithm 1 possesses the advantages of both FCIA and FBIA. We will see in the next section that the parameter δ will play a critical role in the convergence proof of Algorithm 1.

3. Convergence analysis

In this section, we prove the convergence of Algorithm 1 for the minimization problem (12). To this end, we require that there is a positive number ρ such that for any $f\in\mathbb{R}^n$ the framelet matrix $A\in\mathbb{R}^{m\times n}$ satisfies the inequality

$$\|A_1f\|_2\geqslant\rho\|f\|_2, \tag{16}$$

where $A=[A_0^\top,A_1^\top]^\top$ with $A_0\in\mathbb{R}^{m_0\times n}$ and $A_1\in\mathbb{R}^{m_1\times n}$ formed, respectively, from the low-pass filter and the high-pass filters of a tight framelet system. We remark that the matrix A derived from a spline tight framelet system with symmetric boundary extension satisfies the inequality (16) (see, e.g., [5,6]). As a result of (16), we can show the existence of a minimizer to the minimization problem (12). This is stated as follows.

Proposition 3.1. *For a framelet matrix $A\in\mathbb{R}^{m\times n}$ satisfying the inequality* (16) *and a vector $\lambda\in\mathbb{R}^m$ having $\lambda_i\geqslant 0$ for $i=1,\ldots,m_0$ and $\lambda_i>0$ for $i=m_0+1,\ldots,m$, the minimization problem* (12) *has at least one minimizer.*

Proof. To prove the existence of a solution to problem (12), it suffices to show that $\|\mathrm{diag}(\lambda)(Du+c)\|_1$ is convex and coercive with respect to u. The convexity of $\|\mathrm{diag}(\lambda)(Du+c)\|_1$ is a direct consequence of the triangle inequality. Next, we show that $\|\mathrm{diag}(\lambda)(Du+c)\|_1$ is coercive.

Let $\lambda_{\min} := \min\{\lambda_i\colon\ i = m_0+1, m_0+2, \dots, m\}$. For any $u \in \mathbb{R}^{|\Omega\setminus\Lambda|}$, we define a vector $v \in \mathbb{R}^n$ such that $v_{|\Omega\setminus\Lambda} := u$ and $v_{|\Lambda} := g_{|\Lambda}$. Recall that $D = A_{|\Omega\setminus\Lambda}$ and $c = A_{|\Lambda} g_{|\Lambda}$. Then we have that $Av = Du + c$ and

$$\big\|\mathrm{diag}(\lambda)(Du+c)\big\|_1 \geqslant \lambda_{\min}\left(\sum_{i=1}^{m_1}\big|(Du+c)_{m_0+i}\big|^2\right)^{\frac{1}{2}} = \lambda_{\min}\|A_1 v\|_2,$$

which together with the inequality (16) yields $\|\mathrm{diag}(\lambda)(Du+c)\|_1 \geqslant \rho\lambda_{\min}\|v\|_2 \geqslant \rho\lambda_{\min}\|u\|_2$. Clearly, if $\|u\|_2 \to \infty$, then $\|\mathrm{diag}(\lambda)(Du+c)\|_1 \to \infty$ which completes the proof. □

Next, we prove the convergence theorem for Algorithm 1.

Theorem 3.2. *Let $A \in \mathbb{R}^{m\times n}$ be a tight framelet matrix satisfying the inequality* (16) *and $\lambda \in \mathbb{R}^m$ a vector having $\lambda_i \geqslant 0$ for $i = 1, \dots, m_0$ and $\lambda_i > 0$ for $i = m_0+1, \dots, m$. Let $u^\star$ be a solution to problem* (12)*. For $\gamma > 0$ and $0 < \delta \leqslant 1$, then the sequence $\{u^k\}$ generated by Algorithm* 1 *satisfies*

$$\lim_{k\to\infty}\big\|\mathrm{diag}(\lambda)\big(Du^k+c\big)\big\|_1 = \big\|\mathrm{diag}(\lambda)\big(Du^\star+c\big)\big\|_1. \tag{17}$$

Furthermore, when $u^\star$ is the unique solution of problem (12)*, then*

$$\lim_{k\to\infty}\big\|u^k - u^\star\big\|_2 = 0. \tag{18}$$

Proof. For the given vector $\lambda \in \mathbb{R}^m$, we define $E : \mathbb{R}^m \to \mathbb{R}$ for any $d \in \mathbb{R}^m$ as

$$E(d) := \big\|\mathrm{diag}(\lambda)d\big\|_1.$$

With this notation, the minimization problem (12) becomes $\min_u\{E(Du+c)\colon\ u \in \mathbb{R}^{|\Omega\setminus\Lambda|}\}$ whose solutions form a nonempty set by Proposition 3.1.

Let the sequences u^k, d^k, and b^k, $k = 0, 1, \dots$, be generated by Algorithm 1. The three steps in Algorithm 1 can be rewritten as follows:

$$\begin{cases} d^{k+1} = \arg\min\limits_d\left\{\dfrac{\gamma}{2}\big\|Du^k + c - d + b^k\big\|_2^2 + E(d)\colon\ d \in \mathbb{R}^m\right\},\\ u^{k+1} = \arg\min\limits_u\left\{\dfrac{\gamma}{2}\big\|Du + c - d^{k+1} + b^k\big\|_2^2\colon\ u \in \mathbb{R}^{|\Omega\setminus\Lambda|}\right\},\\ b^{k+1} = b^k + \delta\big(Du^{k+1} + c - d^{k+1}\big). \end{cases} \tag{19}$$

Applying the first order optimality condition for the first two equations in (19) leads to

$$\begin{cases} 0 = p^{k+1} + \gamma\big(d^{k+1} - \big(Du^k + c + b^k\big)\big),\\ 0 = \gamma D^\top\big(Du^{k+1} + c - d^{k+1} + b^k\big) \end{cases} \tag{20}$$

with some $p^{k+1} \in \partial E(d^{k+1})$.

Let $u^\star$ denote an arbitrary solution of the model $\min_u\{E(Du+c)\colon\ u \in \mathbb{R}^{|\Omega\setminus\Lambda|}\}$. Define $d^\star := Du^\star + c$. By the first order optimality condition in convex analysis, there exists a vector $p^\star \in \partial E(d^\star)$ such that

$$0 = D^\top p^\star. \tag{21}$$

Due to $E(Du^k + c) \leqslant E(Du^k + c - d^{k+1}) + E(d^{k+1})$ by the triangle inequality and $0 = \langle u^k - u^\star, D^\top p^\star\rangle = \langle Du^k - Du^\star, p^\star\rangle = \langle Du^k + c - d^{k+1}, p^\star\rangle + \langle d^{k+1} - d^\star, p^\star\rangle$ by (21), we obtain

$$E\big(Du^k+c\big) - E\big(Du^\star+c\big) \leqslant E\big(Du^k + c - d^{k+1}\big) + \big\langle Du^k + c - d^{k+1}, p^\star\big\rangle + B_E^{p^\star}\big(d^{k+1}, d^\star\big). \tag{22}$$

Hence, to prove (17), it suffices to show that

$$\lim_{k\to\infty} B_E^{p^\star}\big(d^k, d^\star\big) = 0 \quad\text{and}\quad \lim_{k\to\infty}\big\|Du^k + c - d^{k+1}\big\|_2 = 0, \tag{23}$$

which will be verified in the following.

Denote $b^\star := \frac{1}{\gamma} p^\star$. Since $d^\star = Du^\star + c$, it can be directly checked that

$$\begin{cases} 0 = p^\star + \gamma\big[d^\star - \big(Du^\star + c + b^\star\big)\big], \\ 0 = \gamma D^\top\big(Du^\star + c - d^\star + b^\star\big), \\ b^\star = b^\star + \delta\big(Du^\star + c - d^\star\big). \end{cases} \tag{24}$$

Let us define

$$u_e^k := u^k - u^\star, \qquad d_e^k := d^k - d^\star, \qquad b_e^k := b^k - b^\star, \qquad p_e^k := p^k - p^\star.$$

Subtracting the first equation of (20) from that of (24) leads to

$$p_e^{k+1} + \gamma\big(d_e^{k+1} - Du_e^k - b_e^k\big) = 0.$$

Therefore,

$$\big\langle p_e^{k+1}, d_e^{k+1}\big\rangle + \gamma\big(\big\|d_e^{k+1}\big\|_2^2 - \big\langle Du_e^k, d_e^{k+1}\big\rangle - \big\langle b_e^k, d_e^{k+1}\big\rangle\big) = 0. \tag{25}$$

Likewise, the second equations of (20) and (24) yield

$$\gamma\big(\big\|Du_e^{k+1}\big\|_2^2 - \big\langle D^\top d_e^{k+1}, u_e^{k+1}\big\rangle + \big\langle D^\top b_e^k, u_e^{k+1}\big\rangle\big) = 0. \tag{26}$$

By adding (26) and (25) together, we have that

$$0 = \big\langle p_e^{k+1}, d_e^{k+1}\big\rangle + \gamma\big(\big\|Du_e^{k+1}\big\|_2^2 + \big\|d_e^{k+1}\big\|_2^2 - \big\langle D^\top d_e^{k+1}, u_e^k + u_e^{k+1}\big\rangle + \big\langle b_e^k, Du_e^{k+1} - d_e^{k+1}\big\rangle\big). \tag{27}$$

Furthermore, subtracting the third equation of (20) from that of (24) yields $b_e^{k+1} = b_e^k + \delta(Du_e^{k+1} - d_e^{k+1})$. Thus, $\|b_e^{k+1}\|_2^2 = \|b_e^k + \delta(Du_e^{k+1} - d_e^{k+1})\|_2^2$. For this equation, by expanding out the term on right-hand side, we have that

$$\big\langle b_e^k, Du_e^{k+1} - d_e^{k+1}\big\rangle = \frac{1}{2\delta}\big(\big\|b_e^{k+1}\big\|_2^2 - \big\|b_e^k\big\|_2^2\big) - \frac{\delta}{2}\big\|Du_e^{k+1} - d_e^{k+1}\big\|_2^2. \tag{28}$$

We further know that

$$\begin{aligned} &\big\|Du_e^{k+1}\big\|_2^2 + \big\|d_e^{k+1}\big\|_2^2 - \big\langle D^\top d_e^{k+1}, u_e^k + u_e^{k+1}\big\rangle \\ &\quad = \frac{1}{2}\big(\big\|Du_e^{k+1}\big\|_2^2 - \big\|Du_e^k\big\|_2^2 + \big\|Du_e^{k+1} - d_e^{k+1}\big\|_2^2 - \big\|Du_e^k - d_e^{k+1}\big\|_2^2\big). \end{aligned} \tag{29}$$

Substituting $\langle b_e^k, Du_e^{k+1} - d_e^{k+1}\rangle$ and $\|Du_e^{k+1}\|_2^2 + \|d_e^{k+1}\|_2^2 - \langle D^\top d_e^{k+1}, u_e^k + u_e^{k+1}\rangle$ in (27) by the expressions on the right-hand sides of (28) and (29), respectively, and summing the resulting equation for k from 0 to K, lead to

$$\begin{aligned} &\frac{\gamma}{2\delta}\big(\big\|b_e^0\big\|_2^2 - \big\|b_e^{K+1}\big\|_2^2\big) \\ &\quad = \sum_{k=0}^{K}\big\langle p_e^{k+1}, d_e^{k+1}\big\rangle + \frac{\gamma}{2}\sum_{k=0}^{K}\big((1-\delta)\big\|Du_e^{k+1} - d_e^{k+1}\big\|_2^2 + \big\|Du_e^k - d_e^{k+1}\big\|_2^2\big) + \frac{\gamma}{2}\big(\big\|Du_e^{K+1}\big\|_2^2 - \big\|Du_e^0\big\|_2^2\big). \end{aligned}$$

Because of $0 < \delta \leqslant 1$, we have that

$$\frac{\gamma}{2\delta}\big\|b_e^0\big\|_2^2 + \frac{\gamma}{2}\big\|Du_e^0\big\|_2^2 \geqslant \sum_{k=0}^{K}\big\langle p_e^{k+1}, d_e^{k+1}\big\rangle + \frac{\gamma}{2}\sum_{k=0}^{K}\big\|Du_e^{k+1} - d_e^k\big\|_2^2. \tag{30}$$

Since the Bregman distance is nonnegative and $\langle p_e^{k+1}, d_e^{k+1}\rangle = B_E^{p^{k+1}}(d^\star, d^{k+1}) + B_E^{p^\star}(d^{k+1}, d^\star)$, where $p^{k+1} \in \partial E(d^{k+1})$ and $p^* \in \partial E(d^\star)$, then $\sum_{k=0}^{\infty}\langle p_e^{k+1}, d_e^{k+1}\rangle$ is a series with nonnegative terms and hence is convergent by (30). The convergence of the series implies that $\lim_{k\to\infty}\langle p_e^k, d_e^k\rangle = 0$. Therefore, $\lim_{k\to\infty} B_E^{p^\star}(d^k, d^\star) = 0$ which is the first equation in (23).

Eq. (30) also asserts that $\sum_{k=0}^{\infty}\|Du_e^k - d_e^{k+1}\|_2^2 < \infty$. Hence, $\lim_{k\to\infty}\|Du_e^k - d_e^{k+1}\|_2 = 0$. By $Du^\star + c = d^\star$, we have that $Du^k + c - d^{k+1} = Du_e^k - d_e^{k+1}$. Thus, $\lim_{k\to\infty}\|Du^k + c - d^{k+1}\|_2 = 0$ which is the second equation in (23). Hence, Eq. (17) holds.

Next, we show that (18) holds when $u^\star$ is the unique solution to (12). In fact, Eq. (17) implies that the sequence $\{u^k\}$ is bounded due to the coerciveness of the function $E(D\cdot + c)$. Therefore, the uniqueness of the solution $u^\star$ to (12) leads to a fact that any convergent subsequence of $\{u^k\}$ must converge to $u^\star$. Hence, (18) holds. □

In summary, by Proposition 3.1, the optimization problem (12) has at least one solution. By Theorem 3.2, for a sequence $\{u^k\}$ generated by Algorithm 1 the sequence of values $\|\mathrm{diag}(\lambda)(Du^k + c)\|_1$ converges to the minimum value of $\|\mathrm{diag}(\lambda)(Du + c)\|_1$ on $\mathbb{R}^{|\Omega\setminus\Lambda|}$. Furthermore, if the solution to problem (12) is unique, the sequence $\{u^k\}$ converges to this unique solution. We remark that the proof of Theorem 3.2 was motivated by the work in [9,17].

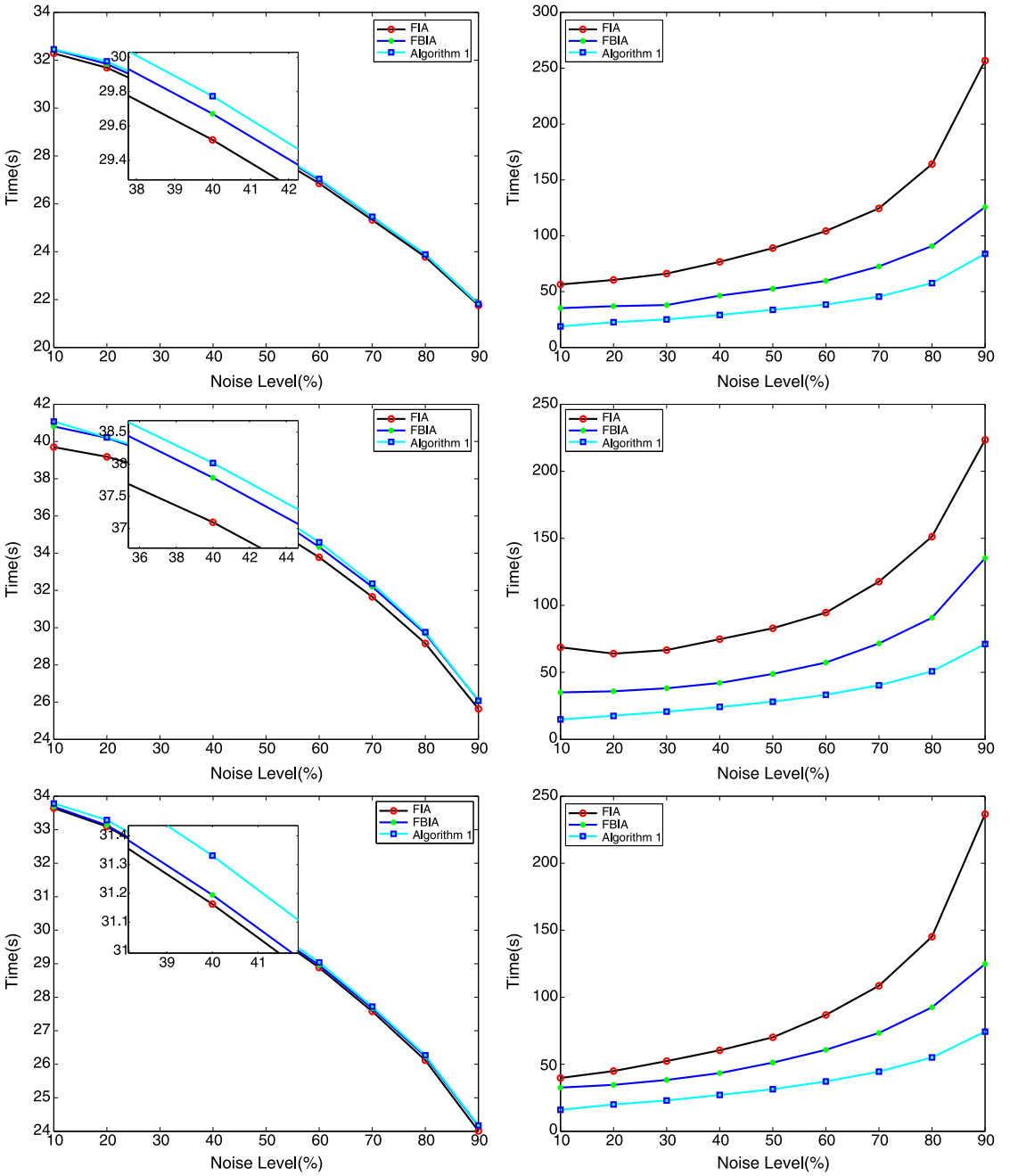


Fig. 1. Performance of the FIA, the FBIA, and Algorithm 1 on the images of "House", "Cameraman", "Lena" corrupted with different level of salt–pepper noise. Column 1: The PSNR-values against noise levels; Column 2: CPU-times against noise levels.

4. Numerical experiments

In this section, we present numerical results of our proposed algorithm. Specially, we will compare the computational performance of the proposed Algorithm 1 with that of FIA and FBIA. All the experiments are performed under Windows 7 and MATLAB R2010a running on a PC equipped with an Intel Core 2 Quad CPU at 3.00 GHz and 2G RAM memory.

We use the "House", "Cameraman", and "Lena" images of size 256×256 as original images in our numerical tests. The objective quality of the restored images is evaluated quantitatively by the peak signal-to-noise ratio (PSNR) which is defined as follows:

$$\mathrm{PSNR} = 20 \log_{10} \frac{255\sqrt{n}}{\|f - f^{\star}\|_2},$$

Table 1
Restoration results of the FIA, the FBIA, and Algorithm 1 with noise level $r = 10\%$, 30%, 50%, 70%, and 90%.

Level	Method	House		Cameraman		Lena	
		PSNR (dB)	Time (s)	PSNR (dB)	Time (s)	PSNR (dB)	Time (s)
10%	FIA	39.70	68.67	32.29	56.37	33.64	39.75
	FBIA	40.81	35.02	32.44	35.15	33.69	32.62
	Algorithm 1	41.07	14.90	32.45	18.83	33.78	16.01
30%	FIA	38.39	66.63	30.68	66.13	32.21	52.37
	FBIA	39.20	39.13	30.85	37.91	32.22	38.39
	Algorithm 1	39.39	20.71	30.96	25.09	32.38	23.00
50%	FIA	35.57	82.91	28.18	88.92	30.00	70.11
	FBIA	36.24	48.81	28.35	52.58	30.04	51.23
	Algorithm 1	36.46	28.05	28.40	33.70	30.14	31.39
70%	FIA	31.66	117.65	25.33	124.55	27.58	108.54
	FBIA	32.19	71.55	25.45	72.57	27.67	73.34
	Algorithm 1	32.37	40.29	25.46	45.47	27.72	44.49
90%	FIA	25.64	223.50	21.76	256.81	24.01	236.58
	FBIA	26.09	135.24	21.82	125.66	24.16	124.84
	Algorithm 1	26.07	71.09	21.81	83.80	24.17	74.29

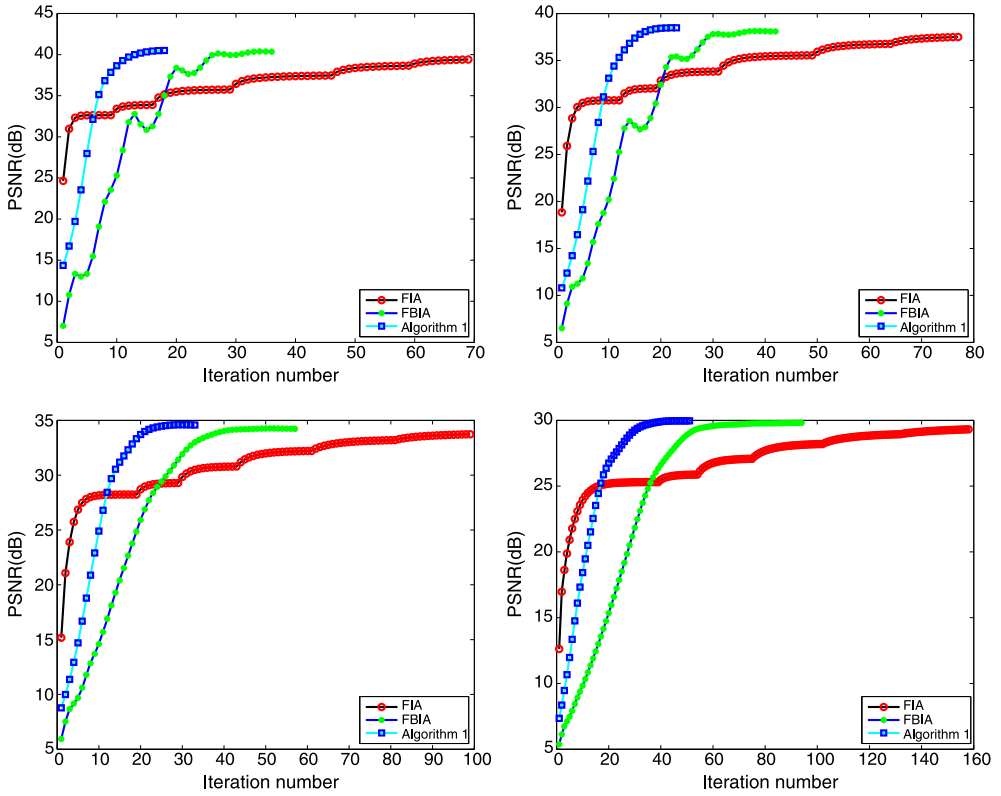


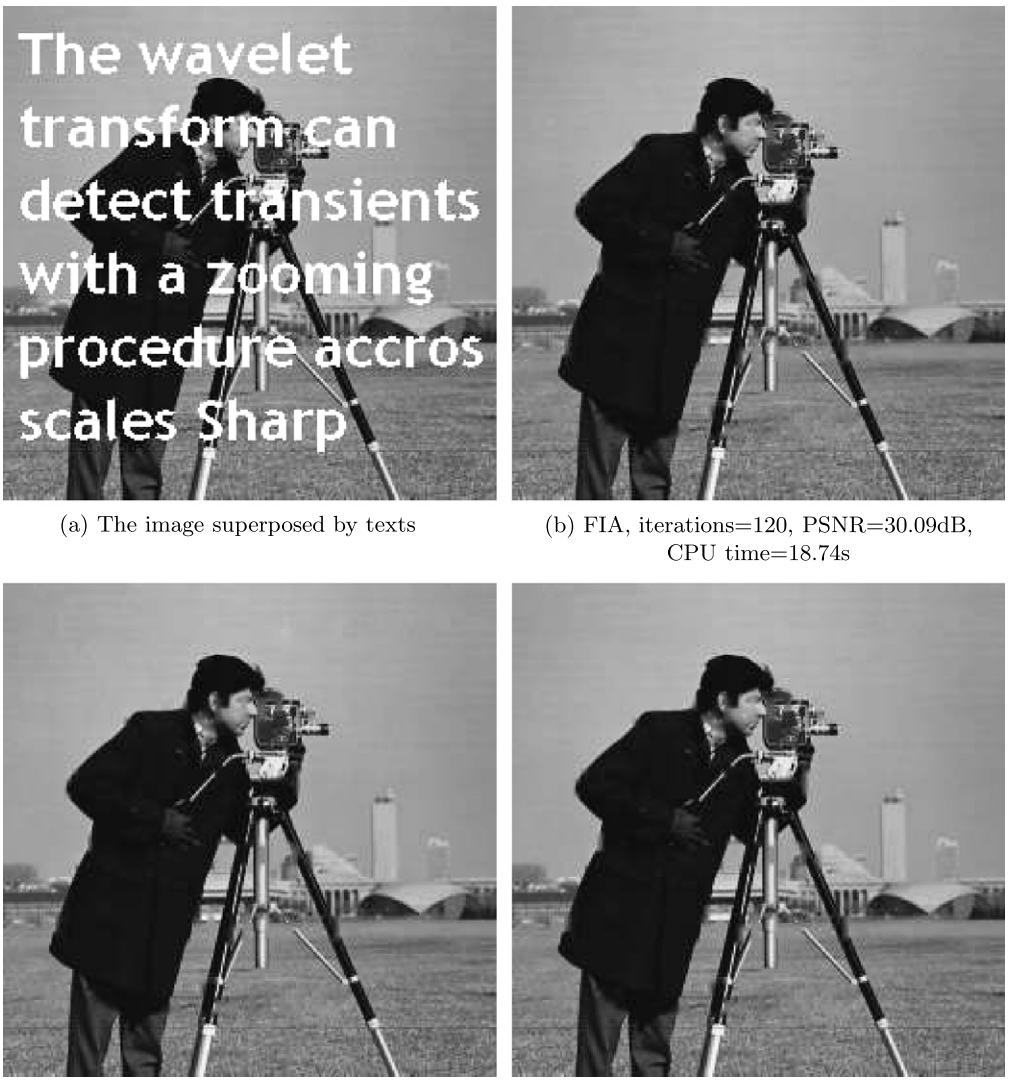
Fig. 2. PSNR history of Algorithm 1, the FIA, and the FBIA for the test image "House" corrupted with different salt–pepper noise level: (top-left) 20% salt–pepper noise, (top-right) 40% salt–pepper noise, (bottom-left) 60% salt–pepper noise, (bottom-right) 80% salt–pepper noise.

with f being the original image, $f^\star$ the restored image, and n the number of pixels in f.

In all numerical experiments, Algorithm 1 with the inpainting parameters $\gamma = 0.2$ and $\delta = 0.5$ is stopped when the relative error between the successive iterates of the restored images satisfies

$$\frac{\|u^{k+1} - u^k\|_2}{\|u^{k+1}\|_2} \leqslant 0.001,$$

where u^k is the denoised image at the k-th iteration. For the FIA and the FBIA, the parameters are determined experimentally for the restored image to achieve the best possible PSNR-value.



(a) The image superposed by texts

(b) FIA, iterations=120, PSNR=30.09dB, CPU time=18.74s

(c) FBIA, iterations=79, PSNR=29.62dB CPU time=14.21s.

(d) Algorithm1, iterations=62, PSNR=30.16dB CPU time=11.14s

Fig. 3. The inpainted images for the image "Cameraman" superposed by texts.

In our first experiment, the framelet matrix A we used is from the piecewise cubic tight framelet system with decomposition level L. The filters associated with the piecewise cubic tight framelet system (see [19]) are $h_0 = \frac{1}{16}[1, 4, 6, 4, 1]$, $h_1 = \frac{1}{8}[1, 2, 0, -2, -1]$, $h_2 = \frac{\sqrt{6}}{16}[-1, 0, 2, 0, -1]$, $h_3 = \frac{1}{8}[-1, 2, 0, -2, 1]$, and $h_4 = \frac{1}{16}[1, -4, 6, -4, 1]$, where h_0 is the low-pass filter and h_k, $1 \leqslant k \leqslant 4$, are the high-pass filters. The detail on how to formulate A from the tight framelet filters can be found in [5,6]. In this case, A is a $m \times n$ matrix with $m = n(1 + 24L)$. The thresholding vector $\lambda \in \mathbb{R}^m$ is chosen as

$$\lambda = \big(\underbrace{0, \ldots, 0}_{n}, \underbrace{2^{-L/2}, \ldots, 2^{-L/2}}_{24n}, \ldots, \underbrace{2^{-l/2}, \ldots, 2^{-l/2}}_{24n}, \ldots, \underbrace{2^{-1/2}, \ldots, 2^{-1/2}}_{24n}\big)^{\top}, \tag{31}$$

as suggested in [7]. Hence the low-pass tight frame coefficients are not thresholded in our Algorithm 1, which is a standard practice in image denoising.

The inpainting problem in the first experiment is to recover images corrupted by salt–pepper noise. In our simulation, the test images were corrupted with "pepper" (pixel value 0) and "salt" (pixel value 255) noise with noise levels varying from 10% to 90%. The decomposition level L is chosen to be 1. In order to use Algorithm 1, the FIA, and the FBIA, the adaptive median filter (AMF) [16] is employed to detect and label the noisy pixels, therefore, generating the inpainting area $\Omega \setminus \Lambda$. The numerical results of noise level 10%, 30%, 50%, 70%, and 90% are listed in Table 1. We found that almost all the PSNR-values of the restored images by Algorithm 1 are higher than those of the FIA and the FBIA. Table 1 also contains the CPU-time consumed by these three algorithms. It is clearly that Algorithm 1 uses much less CPU-time than the FIA and

the FBIA. To better visualize the performance of the three algorithms, the results of the noise level varying from 10% to 90% are plotted in Fig. 1.

To further investigate the convergence properties of the FIA, the FBIA, and Algorithm 1, in Fig. 2 we plot the PSNR-values against iteration numbers of these three algorithms when restoring the test image "House" contaminated by different levels of salt–pepper noise. We observe that PSNR history of the FIA has severe stalling effect. Each time the parameter λ of the FIA decreases, the PSNR-value jumps up quickly for the first several iteration, but soon tends to be stable. As a result, it takes more number of iterations for the FIA to produce the same PSNR-value with the FBIA and Algorithm 1. For the FBIA, when the noise level is below 50%, the PSNR-value has several oscillations; when the noise level is above 50%, the PSNR history of the FBIA is similar to that of Algorithm 1. At all the noise levels, the PSNR-value of Algorithm 1 gets to a stable value much more quickly than the FBIA.

In our second experiment, the considered inpainting problem is to restore images overlying with texts. In Fig. 3(a), the locations of the words beginning with "The" serve as the set $\Omega \setminus \Lambda$. The framelet matrix A in this experiment is from the piecewise linear tight framelet system with the decomposition level L. The corresponding filters are $h_0 = \frac{1}{4}[1, 2, 1]$, $h_1 = \frac{\sqrt{2}}{4}[1, 0, -1]$, and $h_2 = \frac{1}{4}[-1, 2, -1]$. In this case, A is an $m \times n$ matrix with $m = n(1 + 8L)$. The thresholding vector $\lambda \in \mathbb{R}^m$ is chosen as

$$\lambda = \big(\underbrace{0, \ldots, 0}_{n}, \underbrace{2^{-L/2}, \ldots, 2^{-L/2}}_{8n}, \ldots, \underbrace{2^{-l/2}, \ldots, 2^{-l/2}}_{8n}, \ldots, \underbrace{2^{-1/2}, \ldots, 2^{-1/2}}_{8n}\big)^{\top}. \tag{32}$$

The results produced by the FIA, the FBIA, Algorithm 1 with the decomposition level L being 1 are shown in Figs. 3(b)–(d), respectively. One can see that all algorithms can efficiently remove the superposed words. However, Algorithm 1 uses relatively less CPU-time than the FIA and FBIA.

5. Conclusion

In this paper, we propose a new framelet based iterative inpainting algorithm which is derived by using the split-Bregman method. The convergence of this algorithm is given. Numerical comparisons with the framelet based inpainting algorithms FIA and FBIA are given to illustrate the advantages of our algorithm. Numerical results presented in this paper confirm that the proposed algorithm perform favorably and takes much less CPU-time than that of the FIA and the FBIA.

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: Proceedings of SIGGRAPH, New Orleans, LA, 2000, pp. 417–424.
[2] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, Simultaneous structure and texture image inpainting, IEEE Trans. Image Process. 12 (2003) 882–889.
[3] L. Borup, R. Gribonval, M. Nielsen, Bi-framelet systems with few vanishing moments characterize Besov spaces, Appl. Comput. Harmon. Anal. 17 (2004) 3–28.
[4] L. Bregman, The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex optimization, USSR Comput. Math. Math. Phys. 7 (1967) 200–217.
[5] J.-F. Cai, R. Chan, L. Shen, Z. Shen, Simultaneously inpainting in image and transformed domains, Numer. Math. 112 (2009) 509–533.
[6] J.-F. Cai, R. Chan, Z. Shen, A framelet-based image inpainting algorithm, Appl. Comput. Harmon. Anal. 24 (2007) 131–149.
[7] J.-F. Cai, R. Chan, L. Shen, Z. Shen, Convergence analysis of tight framelet approach for missing data recovery, Adv. Comput. Math. 31 (2009) 87–113.
[8] J.-F. Cai, S. Osher, Z. Shen, Linearized Bregman iteration for frame based image deblurring, SIAM J. Imaging Sci. 2 (2009) 226–252.
[9] J.-F. Cai, S. Osher, Z. Shen, Split Bregman methods and frame based image restoration, Multiscale Model. Simul. 2 (2009) 337–369.
[10] T. Chan, S.-H. Kang, J. Shen, Euler's elastica and curvature-based image inpainting, SIAM J. Appl. Math. 63 (2002) 564–592.
[11] T. Chan, J. Shen, Mathematical models for local nontexture inpaintings, SIAM J. Appl. Math. 62 (2002) 1019–1043.
[12] T. Chan, J. Shen, Variational image inpainting, Comm. Pure Appl. Math. 58 (2005) 1019–1043.
[13] T. Chan, J. Shen, H.-M. Zhou, Total variation wavelet inpainting, J. Math. Imaging Vision 25 (2006) 107–125.
[14] M. Elad, J.-L. Starck, P. Querre, D. Donoho, Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA), Appl. Comput. Harmon. Anal. 19 (2005) 340–358.
[15] T. Goldstein, S. Osher, The split Bregman method for ℓ^1 regularization problems, SIAM J. Imaging Sci. 2 (2009) 323–343.
[16] R. Gonzalez, R. Woods, Digital Image Processing, Addison–Wesley, Boston, MA, 1993.
[17] R. Jia, H. Zhao, W. Zhao, Convergence analysis of the Bregman method for the variational model of image denoising, Appl. Comput. Harmon. Anal. 27 (2009) 367–379.
[18] S. Osher, M. Burger, D. Goldfarb, J. Xu, W. Yin, An iterative regularization method for total variation-based image restoration, Multiscale Model. Simul. 4 (2005) 460–489.
[19] A. Ron, Z. Shen, Affine system in $L_2(R^d)$: the analysis of the analysis operator, J. Funct. Anal. 148 (1997) 408–447.
[20] W. Yin, S. Osher, D. Goldfarb, J. Darbon, Bregman iterative algorithms for ℓ^1 minimization with applications to compressed sensing, SIAM J. Imaging Sci. 1 (2008) 143–168.